

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



FELHASZNÁLÓI KÉZIKÖNYV

/USER MANUAL/

a

Folytonos Rendszerek Szimulációjára

készült

A N D I S I M

programnyelvhez

Szalke Erzsébet

Tóth Károly

tudományos munkatársak

1973

A kiadásért felelős

Dr. Vámos Tibor

az

MTA Számítástechnikai és Automatizálási

Kutató Intézetének

igazgatója

Ellenőrizte

dr. Somló János

tudományos főmunkatárs

Készült az Országos Műszaki Könyvtár és Dokumentációs Központ házi
sokszorosítójában

F.v.: Janoch Gyula

Tartalomjegyzék

	oldal
Bevezető	5
1. Egyszerű másodrendű lengőtag vizsgálata ANDISIM-mel	7
2. Az ANDISIM szimulátor általános jellemzői	13
3. Feladatok működési /blokk/ vázlatának összeállításánál lényeges ANDISIM tulajdonságok	15
4. Feladatok előkészítése az ANDISIM-mel való futtatása	17
5. Felhasználói szubrutinok beépítése az ANDISIM-be	23
6. Az ANDISIM hibajelzései	26
Függelék	28
Elemkészlet táblázat	43

Tartalom

Bevezető

A digitális szimulációs programnyelvek fejlődését felhasználói oldalról két igény irányította. Egyrészt az, hogy a szabályozástechnikával foglalkozó mérnökök a digitális számítógépek és gépi nyelvek beható ismerete nélkül is megoldhassák feladatukat egy egyszerűen elsajátítható digitális folytonos szimulációs nyelv segítségével. Másrészt, hogy a kutatók egy olyan kényelmes és flexibilis eszközt kapjanak, amellyel az analóg gépnél pontosabb eredményekhez juthatnak feladatuk gyors és hatékony fordítása és futtatása révén.

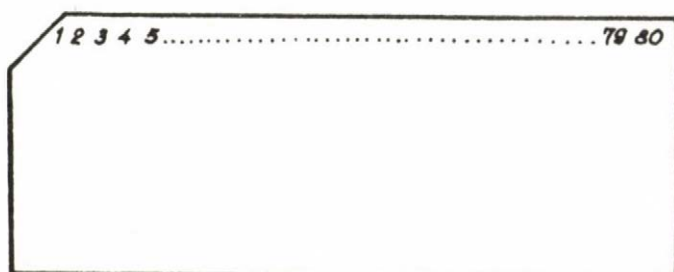
A szélesebb körben elterjedt programnyelvek - mint a MIDAS, a PACTOLUS, a CSMP - általában megfelelnek ezeknek a felhasználói igényeknek.

Mi itt a fenti tulajdonságokkal rendelkező, folytonos rendszerek digitális szimulációjára szolgáló, ANDISIM elnevezésű programnyelvet bocsátjuk a felhasználók rendelkezésére, amely az MTA SZTAKI CDC 3300-as számítógépre adaptált és kifejlesztett szimulátor. Röviden összefoglaltuk ebben a leírásban a használatához szükséges tudnivalókat, melyek egyszerű és gyors megismerését a bemutatott feladatok is elősegítik.

1. Egyszerű másodrendű lengőtag
vizsgálata ANDISIM-mel.

Analóg számítógépen már futott feladaton, a 2. ábrán látható egyszerű másodrendű lengőtag vizsgálatán keresztül adunk először ízelítőt az ANDISIM használatáról.

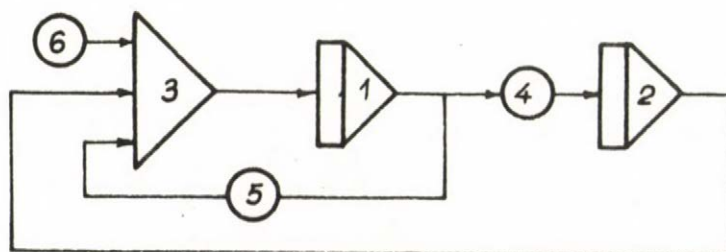
1.1. Az ANDISIM FORTRAN bázisnyelven készült a CDC 3300-as számítógépre, ezért a feladatok bemenő adatait az 1. ábrán vázolt alakú FORTRAN kártyákra kell lelyukasztatni IBM kártyalyukasztón.



1. ábra

1.2. A 4. fejezetben ismertetjük részletesen, hogy a legáltalánosabb esetben milyen adatkártyákat tartalmaz egy futtatásra kész feladat adatkártyacsomója.

Itt most csupán azokat az adatkártyákat soroljuk fel, amelyek az alábbi ábrán látható másodrendű lengőtag vizsgálatához szükségesek.



2. ábra

1.2.1. Az adatkártyák közül az első 11 az ANDISIM-nek a CDC 3300-as számítógépen való aktivizálására szolgáló vezérlő kártya. Az első ilyen vezérlő kártyáról, az un. JOB kártyáról a 4. fejezetben lesz szó részletesen. Ez ugyanis a felhasználótól és feladattól függő információkat hordoz.

A 2.-11. kártyák a felhasználótól és feladattól függetlenül kötelezően az alábbi alakúak:

1.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 80
\$JOB, 22F871, LENG00, 2, 2000, , ,
2.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 80
\$SCHED, CORE=81, SCR=2, CLASS=I, 854=1, OUT=512
3.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 80
\$*DEF(0, , XXX3, 0003, PLOTTERFILE-HEADER, , , 0)
4.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 80
\$*DEF(0, , ANAL, 003333, ANAL, , , I)
5.

1 2 3 4 5 6 7 8 9 10 11 80
\$FILE, 3=INP
6.

1 2 3 4 5 6 7 8 9 10 11 80
\$FILE, 4=OUT
7.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 80
\$*DEF(0, , 7777, UNRZ05, PLOTTERFILE, , MM, 0)
8.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 80
\$*DEF(0, , ALIB, MASTER, LIBRARY, PL, *D0*)
9.

1 2 3 4 5 6 7 8 9 10 11 12 13 80
\$*DEF(0, , ADIR, MASTER, LIBRARY-DIRECTORY, PL, *D0*)
10.

1 2 3 4 5 6 7 8 9 10 80
\$ANAL, ANAL
11.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 80
\$AUX, ALIB, ADIR

1.2.2. Az adatkártyacsomó felhasználótól a feladattól függő adatblokkja a következő típusu kártyákból áll:

a/ Minden ilyen adatblokkot egy un. RUN kártya vezet be és zár le, amely a feladat futására és ennek során igényelt szolgáltatásokra ad direktívákat 8 kulcs értékének beállításával. A kulcsok szerepéről a 4. fejezetben tájékoztatunk részletesen. Itt most a bevezető RUN kártya képét adjuk, amely a 12. kártya az adatkártyacsomóban

12.	1	2	3	4	5	6	7	8	...	80
	1	1	1	1	1	1	0	0		

b/ A blokk-konfigurációt leíró kártyák összeállításához a másodrendű lengőtagnak a 2. ábrán látható analóg blokkvázlatát használjuk fel. Minden egyes blokk sorszámához és a Függelék táblázatából vett típusjelző számához megadjuk a bemenetére kapcsolódó blokkok sorszámát:

	Blokk sorszám	Blokk típus	Első bemenet	Második bemenet	Harmadik bemenet
13.	18, 19, 20, 21	28, 29, 30, 31	38, 39, 40, 41	48, 49, 50, 51	58, 59, 60, 61 ... 80
	1	25	3		
14.	19, 20	29, 30	39, 40	49, 50	59, 60 ... 80
	2	25	4		
15.	19, 20	29, 30	39, 40	49, 50	59, 60 ... 80
	3	54	6	2	5
16.	19, 20	29, 30	39, 40	49, 50	59, 60 ... 80
	4	23	1		
17.	19, 20	29, 30	39, 40	49, 50	59, 60 ... 80
	5	23	1		
18.	19, 20	29, 30	39, 40	49, 50	59, 60 ... 80
	6	34			
19.	üres kártya				

Az utolsó üres kártya a konfigurációleírás végét jelzi.

c/ Paraméterek listáját specifikáló kártyák:

	Blokk sorszám	Első paraméter	Második paraméter	Harmadik paraméter
20.	3	1.0	-1.0	-1.0
21.	4	1.0		
22.	5	1.4		
23.	6	100.0		
24.	üres kártya			

A blank kártya itt is a paraméterspecifikáció végét jelzi.

d/ Az integrálási eljárást, lépéshosszat, intervallumot és nyomtatósi lépésközt megadó kártya:

25.	1	0.05	5.0	0.05
-----	---	------	-----	------

e/ A sornyomtatón regisztrálandó blokkok felsorolása:

26.	3	1	2	4	5
-----	---	---	---	---	---

f/ A plotterre vonatkozó adatok:

27.	7777	1000.	600.
-----	------	-------	------

g/ Most következik az adatblokk végét jelző RUN kártya. Ez, itt látható kártyaképpel arra utal, hogy a feladatot nem kell módosított változatban ujrafuttatni:

28.	00000000
-----	----------

h/ Az egész adatkártyacsomót lezáró un. end-of-file kártya mint vezérlőkártya jelzi a beolvasandó kártyák végét:

29. 1 2 3 ... 80
77
88

1.3. A futás végén kézhezkapott listán a visszaírt adatokat, a számszerű eredményeket és a printeren kirajzolt görbéket kapjuk meg az alábbi formában: DIGITAL ANALOG SIMULATOR PROGRAM

CONFIGURATION SPECIFICATIONS				
BLOCK	TYPE	INPUT 1	INPUT 2	INPUT 3
1	INTEGRAL	3	0	0
2	INTEGRAL	4	0	0
3	RECT. SLP	6	2	5
4	GAIN	1	0	0
5	GAIN	1	0	0
6	CONSTANT	0	0	0

FUNCTION GENERATOR SPECIFICATIONS

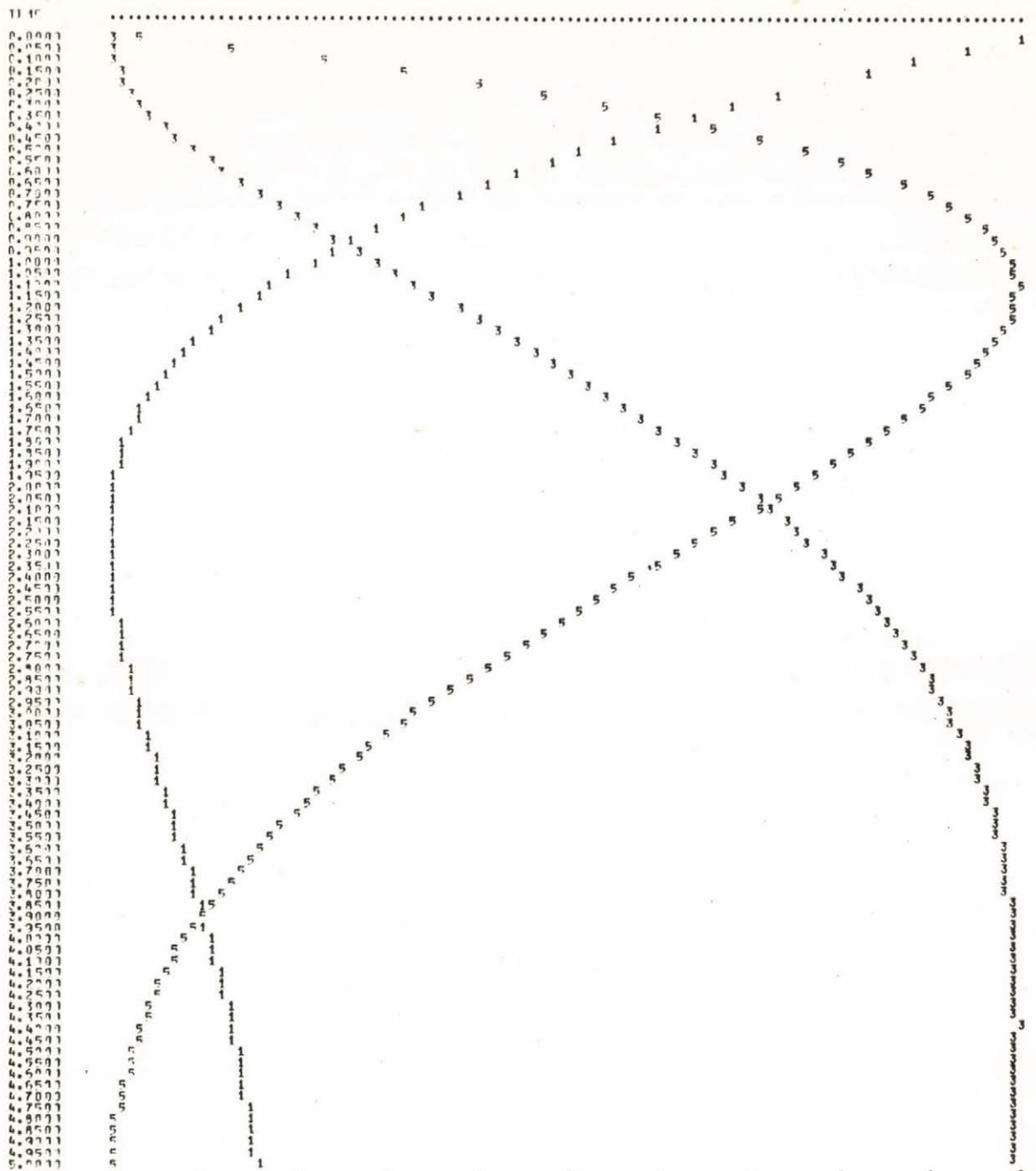
INITIAL CONDITIONS AND PARAMETERS			
BLOCK	IC/PAAR1	PAAR2	PAAR3
3	1.00000	-1.00000	-1.00000
4	1.00000	0.00000	0.00000
5	1.40000	0.00000	0.00000
6	100.00000	0.00000	0.00000

INTEGRATION INTERVAL 0.050000000
 TOTAL TIME 5.000000000
 PRINT INTERVAL 0.050000000
 FIX RK2 INTEGRATION

TIME	DKSTEP	OUTPUT 3	OUTPUT 1	OUTPUT 2	OUTPUT 4	OUTPUT 5
0.00000	0.50000E-01	100.00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.05000	0.50000E-01	93.120	4.8250	0.12500	4.8250	6.7550
0.10000	0.50000E-01	86.481	4.3120	0.48265	9.3120	13.037
0.15000	0.50000E-01	80.081	13.473	1.0564	13.473	18.862
0.20000	0.50000E-01	73.922	17.320	1.8301	17.320	24.248
0.25000	0.50000E-01	68.000	20.865	2.7845	20.865	29.211
0.30000	0.50000E-01	62.315	24.120	3.9156	24.120	33.768
0.35000	0.50000E-01	56.864	27.097	5.2007	27.097	37.935
0.40000	0.50000E-01	51.664	29.807	6.6266	29.807	41.729
0.45000	0.50000E-01	46.653	32.261	8.1815	32.261	45.166
0.50000	0.50000E-01	41.897	34.472	9.8528	34.472	48.260
0.55000	0.50000E-01	37.342	36.450	11.629	36.450	51.030
0.60000	0.50000E-01	32.914	38.206	13.498	38.206	53.488
0.65000	0.50000E-01	28.609	39.751	15.450	39.751	55.651
0.70000	0.50000E-01	24.403	41.096	17.473	41.096	57.534
0.75000	0.50000E-01	21.200	42.250	19.559	42.250	59.150
0.80000	0.50000E-01	17.707	43.225	21.694	43.225	60.515
0.85000	0.50000E-01	14.478	44.020	23.882	44.020	61.640
0.90000	0.50000E-01	11.357	44.672	26.101	44.672	62.541
0.95000	0.50000E-01	8.4234	45.165	28.344	45.165	63.230
1.00000	0.50000E-01	5.6613	45.514	30.618	45.514	63.720
1.05000	0.50000E-01	3.0763	45.731	32.901	45.731	64.023
1.10000	0.50000E-01	0.6015	45.822	35.191	45.822	64.151
1.15000	0.50000E-01	-1.5000	45.796	37.483	45.796	64.115
1.20000	0.50000E-01	-3.6074	45.662	39.771	45.662	63.927
1.25000	0.50000E-01	-5.6466	45.427	42.049	45.427	63.597
1.30000	0.50000E-01	-7.4400	45.197	44.314	45.197	63.136
...

PLOTTED OUTPUTS

3.	HL	OCKSYH90L	1
1.	HL	OCKSYH90L	2
2.	HL	OCKSYH90L	3
4.	HL	OCKSYH90L	4
5.	HL	OCKSYH90L	5



3. RLK	-21.0	-8.91	7.24	15.3	27.4	39.5	51.6	63.7	75.8	87.9	100.
1. RLK	-1.75	3.01	7.76	12.5	17.3	22.0	26.8	31.5	36.3	41.1	45.8
2. RLK	0.000000	10.5	20.9	31.4	41.8	52.3	62.8	73.2	83.7	94.1	105.
4. RLK	-1.75	3.01	7.76	12.5	17.3	22.0	26.8	31.5	36.3	41.1	45.8
5. RLK	-2.45	4.21	17.9	17.5	24.2	30.8	37.5	44.2	50.8	57.5	64.2

LENGTH OF X AXIS 100.
LENGTH OF Y AXIS 500.

Y SCALE FACTORS FOR PLOTTED OUTPUT:
 $0.495887E+11$ $1.125122E+02$ $0.577637E+01$ $1.125122E+02$ $0.31947E+01$

2. Az ANDISIM szimulátor általános jellemzői.

- 2.1. Az ANDISIM folytonos rendszerek szimulációjára szolgáló blokkorientált programnyelv.
- 2.2. A standard analóg elemeken kívül összehasonlító, logikai, késleltető elemeket is tartalmaz. Implicit függvény elemei elvégzik a konvergens algebrai hurkok feldolgozását. A felhasználónak lehetősége van további kilenc, tetszése szerinti elem beépítésére, ami jelentősen szélesíti a megoldható feladatok körét. Ilyen speciális elemek beépítésének módjáról az 5. fejezet tájékoztat részletesen. A jelenlegi elemkészletet a Függelék I. táblázata tartalmazza.
- 2.3. A feladatok futtatása gyorsá és hatékonyá vált az ANDISIM segítségével, ugyanis egyszeri adatbeolvasással és fordítással lehet vele egy feladatot több változatban újrafuttatni. Így lehetőséget ad a konfiguráció, a paraméterértékek, az integrálási eljárás adatainak változtatására, valamint az eredményközléssel kapcsolatos felhasználói igények módosítására egy futáson belül. Mindezt a felhasználó az 5. mintafeladatban láthatja közelebbről.
- 2.4. Három, különböző pontosságú integrálási eljárás áll a felhasználó rendelkezésére az ANDISIM használatakor. Gyors eredményre vezet a fixlépéses másodrendű, illetve negyedrendű Runge-Kutta módszer. Minden esetben - instabil feladatoknál is - megbízható eredményt ad az automatikusan váltott lépéshosszu negyedrendű Kutta-Gill eljárás.
- 2.5. A feladatok bemenő /input/ adatait előzetesen összeállított és lelyukasztott adatkártya csomóról olvastatják be a CDC 3300 kártyaolvasóján, Az input adatok összeállításához a 3. és 4. fejezet ad utmutatást.
- 2.6. A felhasználó által az adatkártyákon megjelölt regisztrátumok /output eredmények/ számszerűen kinyomtatva és kirajzolva nyerhetők a feladat futása során a CDC 3300 sornyomtatóján. Szükség esetén a görbék plotte-re is kirajzoltathatók.

2.7. A szimulátor automatikus szortoló rutinnal rendelkezik, amely egy adott feladatnál szereplő blokkokat tényleges működésük sorrendjébe állít. Ugyanis a digitális számítógép soros működése következtében a szimulációs konfiguráció által előírt számítások minden időciklusban meghatározott sorrendben kerülhetnek csak végrehajtásra. Ez a sorrend arra a kritériumra épül, hogy csak olyan blokkra lehet a számítást elkezdeni, amelynek valamennyi bemeneti értéke ki van számítva.

3. Feladatok működési /blokk/
vázlatának összeállításánál lényeges
ANDISIM tulajdonságok.

- 3.1. Az ANDISIM blokkorientált szimulátor, tehát használatához - ugyanugy, mint az analóg számítógépes megoldáshoz - először a feladat blokkvázlatát, vagy más néven működési vázlatát kell összeállítani. Ennél a Függelék elemkészlet-táblázatának elemei használhatók fel. Az ANDISIM egy blokksémában max.75 elem egyidejű használatát engedi meg. Ezek közül egy feltétlenül integrátor kell hogy legyen. Egy blokkkonfigurációban integrátorból max 25, késleltető elemből max.5 és függvénygenerátorból is max.5 szerepelhet.
- 3.2. Már a blokkséma összeállításánál figyelembe kell venni, hogy az ANDISIM elemei max.3 bemenetűek. Összegzői, integrátorai nem előjelfordítók, mint a megfelelő analóg számítógépi elemek, hanem előjeltartók. Arra nézve, hogy az egyes blokkok az elemkészletből hány bemenettel rendelkeznek és működésükhöz hány és milyen rendeltetésű paramétert kell megadni, a Függelék elemkészlet-táblázata ad eligazítást.
- 3.3. Külön felsoroljuk most egyes elemeknek olyan fontos tulajdonságait, amelyeket az elemkészlet-táblázatban nem tüntettük fel, vagy onnan nem nyilvánvalók.
- 3.3.1. A függvénygenerátorok mindegyikére maximum 23 összetartozó abszcissza-ordináta érték adható meg. Ezen alappontok felhasználásával egy függvénygenerátor a bemenetére adott értékhez mint abszcisszához lineáris interpolációval állítja elő a megfelelő ordinátaértéket, amely kimenetén megjelenik.
- 3.3.2. Az integrátorok hárombemenetű elemek. Az első bemenetükre a felhasználó által megadott paramétert a szimulátor úgy kezeli, mint az integrátorra $t=0.0$ időpillanatban megadott kezdeti feltételt. Az első bemenethez tartozó súly-paramétert ugyanakkor konstans módon 1.0-re állítja be. A felhasználó által a másik két bemenethez rendelt paraméterértékeket már a szokásos paraméter súlyokként használja.

Ezekkel és az 1.bemenetben beállított 1.0 paraméterrel a súlyozott összegzőknél megszokott módon áll elő az integrátor eredő bemeneti értéke.

- 3.3.3. Az idő impulzus generátor a bemenő jeltől függően változó időimpulzust állít elő. A bemenő jel pozitivitásának ideje alatt adott P_1 periódussal hoz létre egységimpulzust egy lépéshossznak megfelelő időre. Működését a Függelék 1. mintafeladatán keresztül mutatjuk be.
- 3.3.4. A delay vagy késleltető elemek a bemenő jelet $n \cdot h$ idővel késleltetve adják ki, ahol n az 1,2,...,50 egészszámok valamelyike, h pedig az integrálási lépéshossz. Ha egy blokk-konfigurációban több késleltető is előfordul, ezeknél a késleltetés mértéke egymástól függetlenül különböző lehet.
- 3.3.5. Konvergens algebrai hurok kezelésére, vagyis az $Y=f(Y,X)$ implicit függvény értékeinek iterációval történő számítására a Wye és Vacuous elnevezésű implicit függvény elemek szolgálnak. Ezek mindig együtt kell hogy szerepeljenek egy működési vázlatban. A Vacuous elem az iteráció indításához szükséges kezdeti Y értéket, s minden további iterációhoz az előbbi iterációval nyert Y értéket szolgáltatja. A Wye elem az Y és az $f(Y,X)$ értékek összehasonlítását végzi. Míg ezek relatív eltérése egy megadott értéknél nagyobb, folytatódik az iteráció. Ha kisebb, vagy egyenlő a megadott értékkel, akkor az iteráció végetér. Ekkor kerül csak sor újból a blokksémában szereplő többi elem működésére, amely az iteráció kezdetével fel lett függesztve. A Wye és Vacuous elemek funkcióját a Függelék 2.mintafeladatában ismerheti meg a felhasználó működés közben. A feladatból az is kitűnik, hogy az Y gyakran csak mint közbülső változó szerepel egy $Z=g(Y,X)$ formájú későbbi egyenletben.

ANDISIM-mel való futtatásra.

4.1. Egész általánosan sorravesszük azokat a lépéseket, amelyeket végigjár a felhasználó feladata futásra való előkészítésekor.

4.1.1. Az analóg számítógépes megoldáshoz hasonlóan itt is először a feladat blokkvázlatát kell összeállítani és a benne szereplő blokkokat 1-től kezdődően folyamatosan sorszámmal ellátni. Célszerű ezután minden egyes blokkhoz mint műveleti elemhez kikeresni az un.típusjelző számot a függelékben található elemkészlet táblázatból. Ugyanis a blokkok sorszáma és típusjelző száma segítségével készítjük el a szimulátor számára lényeges konfiguráció leírást tartalmazó kártyákat. Minden egyes, a sorszámával és típusjelző számával azonosított blokkhoz meg kell adni a blokk bemenetére kapcsolódó elemek sorszámát egy ilyen kártyán:

[illegible]

A konfigurációt specifikáló kártyák sorát egy vezérlési célt szolgáló blank /üres/ kártya zárja le.

4.1.2. Amennyiben van a konfigurációban függvénygenerátor, akkor a konfiguráció után a függvénygenerátorok specifikálása következik.

Az első kártyán mindig a függvénygenerátor pontpárjainak számát adjuk meg:

a függvénygenerátor pontpárjainak száma = 9

pl. 1 2 3 4 ... 80

9

A következő kártyán a függvénygenerátor-blokk sorszáma után az első, maximálisan hét, ordinátaértéket tüntetjük fel:

pl.

a blokk sorszáma	első ordináta	második ordináta	hetedik ordináta
1 2 3 ... 8 9 10 11	20 21	30 31	40 41 ... 50 51 ... 60 61 ... 70 71 ... 80
8	5.0	10.0	

kat tartalmazó kártya következnek:

	első abszcissza	második abszcissza	harmadik abszcissza	hetedik abszcissza
pl.	1 2 3 4 ... 10 11	20 21	30 31	40 41 50 51 60 61 70 71	80
	0.1	0.2			

nyolcadik osztalyra kitenyelek stb.

nucleotide
 difference
 difference
 difference

12 10 11
 20 21
 30 31
 40 41
 50 51
 60 61
 70 71
 80

4.1.3. A következő lényeges adatkártyacsoport a paraméter specifikációt szolgáltatja azokra a blokkokra, amelyeknél paraméter megadásra szükség van. Ezeknél a blokkoknál a sorszám után a három lehetséges paramétert kell feltüntetni a kártyán:

a blokk sorozata = 3, első paraméter = 0.1, második paraméter = 0.8, harmadik paraméter = 0.4

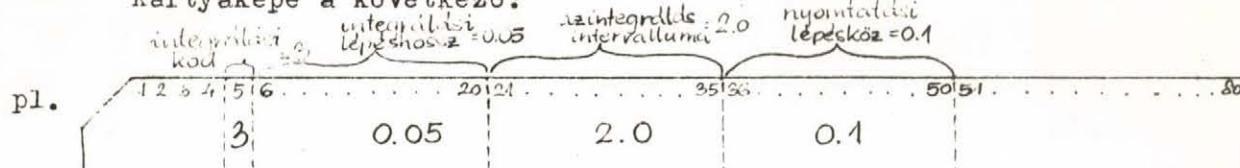
pl.	12	18	19	20	21	35	36	40	41	65	66	80
			3			0.1		0.8		0.4		

4.1.4. Az integrálási eljárásra vonatkozó adatkártyák közül az első kártya 4 adatot tartalmaz:

3 a kód, ha automatikus lépésváltásu negyedrendű Kutta-Gill módszer használunk/;

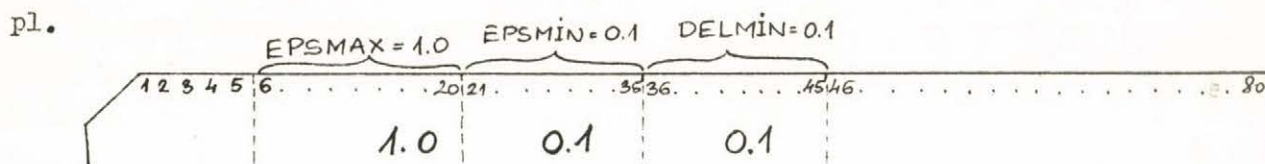
- b/ az integrálás lépéshosszát;
- c/ a teljes integrálás intervallumát és
- d/ a nyomtatások időközét.

Kártyaképe a következő:

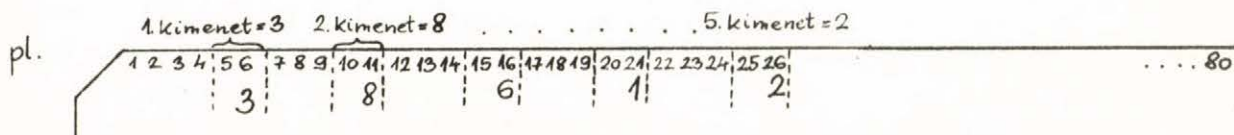


Második kártyára az integrálási specifikációnál csak akkor van szükség ha az automatikus lépéshosszmódosító Kutta-Gill eljárást választottuk, vagyis az előbbi kártyán az integrálási kód 3. Ez a második kártya 3 adatot tartalmaz:

- a/ egy olyan szorzótényezőt, amely azt adja meg, hogy a kezdeti lépéshossz hányszorosánál nagyobb hiba esetén kerüljön sor a lépéshossz lefelezésére /EPSMAX/,
- b/ egy szorzótényezőt, amely megadja, hogy a kezdeti lépéshossz hány-szorosánál kisebb hiba esetén engedjük meg a lépéshossz duplázását /EPSMIN/,
- c/ egy szorzótényezőt, amely azt szabja meg, hogy a kezdeti lépéshossz hány-szorosa lehet minimálisan a lefelezésekkel kapott új lépéshossz /DELMIN/.



4.1.5. A regisztrálandó blokkokat kijelölő kártya a sorszámmal adja meg a blokkokat.



Maximálisan 5 kimenet regisztrálható egyidejűleg.

4.1.6. A plottert és plotter-rajzot előíró kártyán három adat áll.

- a/ a plotterre vonatkozó konstans azonosítószám: 7777
- b/ a plotter-rajz x-tengelymenti mérete /0.01 inchben, azaz plotter

lépésben mérve/

c/ a plotter-rajz y-tengelymenti mérete /0.01 inchben, azaz plotter lépésben mérve/

pl.	1	2	3	4	5	6	...	20	21	...	35	36	...	80
	7	7	7	7				1	0	0	0	0		6
														0

4.2. Itt kell szólnunk a bevezető mintafeladatban már említett RUN-kártyáról. A függelék mintafadatai is mutatják, hogy ilyen kártya fog közre az adatkártyacsomóban minden a 4.1.1-4.1.6 alatti kártyákból álló adatblokkot.

Tehát a sorrend egy adatblokkon belül a következő:

- a/ RUN-kártya
- b/ Konfigurációt specifikáló kártyák /üres kártyával lezárva/
- c/ függvénygenerátorokra vonatkozó kártyák
- d/ paramétereket specifikáló kártyák /az utolsó egy blankkártya/
- e/ az integrálási eljárásra vonatkozó kártyák
- f/ a regisztrátumokat előíró kártya
- g/ a plotterre vonatkozó kártya
- h/ RUN-kártya.

Egy ilyen adatblokk után a végén álló RUN-kártya képétől függően vagy újabb adatblokk következik, amely a feladat újabb futtatásánál érvényes módosított adatokat tartalmazza, vagy pedig a futásnak vége, és az utolsó RUN-kártyát már csak egy "end-of-file" kártya követi a bevezető mintafeladatból ismert kártyaképpel.

A RUN-kártya belső vezérlési feladatot lát el azzal, hogy rajta a felhasználó 8 kulcs értékét állítja 1-re, vagy 0-ra, aszerint, hogy az illető kulccsal irányított működést igényli-e vagy sem:

1. kulcs	2. kulcs	3. kulcs	4. kulcs	5. kulcs	6. kulcs	7. kulcs	8. kulcs	...	80
1	2	3	4	5	6	7	8		

- Az 1 kulcs : a feladat futását
- a 2 -" : a konfiguráció specifikálást

- a 3.kulcs : a függvénygenerátor megadást
- a 4. -"- : a paraméterek beolvastatását
- az 5. -"- : az integrálási eljárás specifikálását és a regisztrál-
landó kimenetek kijelölését
- a 6. -"- : a regisztrátumok plotteren való kirajzoltatását
- a 7. -"- : a regisztrált görbék egymástól független skálázását
- a 8. -"- : a számolásnélküli, módosított skálafaktorral történő
ujabb kirajzoltatást /printeren és plotteren/

irányítja.

A RUN-kártyán beállítható kulcsok segítségével tudjuk generálni a kiindulási feladat újabb futását pl.módosított paraméterekkel, vagy módosított integrálási kondíciókkal, és ilyenkor elegendő a futást bevezető RUN-kártya után csupán a módosítást tartalmazó kártyákat elhelyezni. A

konfiguráción, vagy paramétereken való módosítás kártyái után ebben az újabb adatblokkban is szükséges a blank-kártya, és ezt az újabb adatblokkot is egy RUN-kártya zárja le. Az utolsó RUN-kártya mind a 8 kulcspozícióján 0-t tartalmaz. Ezt már csak az end-of-file kártya követi. Ily módon a feladat annyi változatban futtatható le egyszeri adatbeolvasással és fordítással, amennyi becslésünk szerint a - következőkben ismertetésre kerülő - JOB kártyán feltüntetett futtatási időbe belefér.

4.3.A bevezető mintafeladatnál már utaltunk arra, hogy az adatkártyacsomó elején álló vezérkártyák közül az első, a JOB kártya képe felhasználótól és feladattól függő. Azt is említettük, hogy a JOB kártya és az azt követő 10 vezérkártya arra szolgál, hogy a CDC 3300 számítógép az ezekből és a fentiekben leírt adatblokk kártyáiból álló kártyacsomót elfogadja futásra és ehhez az ANDISIM-et aktivizálja.

A JOB kártya képe egész általánosan a következő:

1 2 3 4

\$JOB,nnnnnn,PRNEV,t,l,,,

. . .80

FELHASZNÁLÓ NEVE

Itt az nnnnnn: egy hatjegyű szám, amelyet bejelentkezésekor kap a felhasználó a CDC számítóközponttól mint számlaszámot,

PRNEV: a feladat neve, legfeljebb 6 karakter,
t : az előzetes becslés szerint a feladat futásához
szükségesnek tartott idő percekben,
l : a program futása során kinyomtatott sorok száma
/legfeljebb 1000 sor/.

5. Felhasználói szubrutinok

beépítése az ANDISIM-be

5.1. A FORTRAN nyelvű programozásban jártas felhasználónak lehetőséget nyújt az ANDISIM szerkezete arra, hogy kilenc tetszés szerinti funkcióju FORTRAN szubrutint építhessen be a szimulátorba. A rendszer ezáltal igen flexibilissé válik, mert sajátos igényeknek megfelelő elemekkel, algoritmusokkal bővíthető a feladatok által megszabott speciális követelmények szerint.

5.2. Egy-egy ilyen FORTRAN forrásnyelvű felhasználói szubrutint a következő szegmensnyitó és záró utasítások közé kell foglalni:

```
SUBROUTINE SUBN(C,PAR,I,J,K,L)
  DIMENSION C(76), PAR(75,3)
  .....
  RETURN
END
```

Egy konkrét felhasználói rutinnál a SUBN szubrutinnév helyén a SUB1, SUB2, ..., SUB9, nevek valamelyike állhat. /Természetesen mindegyik csak egyszer használható fel./

A szubrutin fejben szereplő paraméterek jelentése a következő:

- I bemenő paraméter, a szóban forgó speciális elem blokkazonosító száma, melyet a konfiguráció megadásakor olvastatunk be;
- J bemenő paraméterek, az I sorszámú speciális elem bemenetére
- K kapcsolódó három lehetséges blokk azonosító száma, melyek u-
- L gyancsak a konfigurációleírásban kerülnek beolvasásra I-vel együtt

A C vektor

- C(J) elemei bemenő paraméterek, a J, K illetve L sorszámú blokkra a szimulátor által kiszámított aktuális kimeneti értékek, melyek
- C(K) egyuttal az I-edik speciális elem működéséhez szükséges bemene-
- C(L) ti értékek;

C(I) eleme kimenő paraméter, az I-edik blokk kimeneti értéke, amelyet a speciális elem algoritmusára eredményez;

C(76) eleme bemenő paraméter, a folyó idő aktuális értéke;

PAR(I,1) }
PAR(I,2) } tömbelemek bemenő paraméterek, az I-edik blokk első, második,
PAR(I,3) } illetve harmadik bemenetéhez rendelhető paraméterértékek,
az adatkártyacsomó paramétereit specifikáló kártyáiról be-
olvastatva.

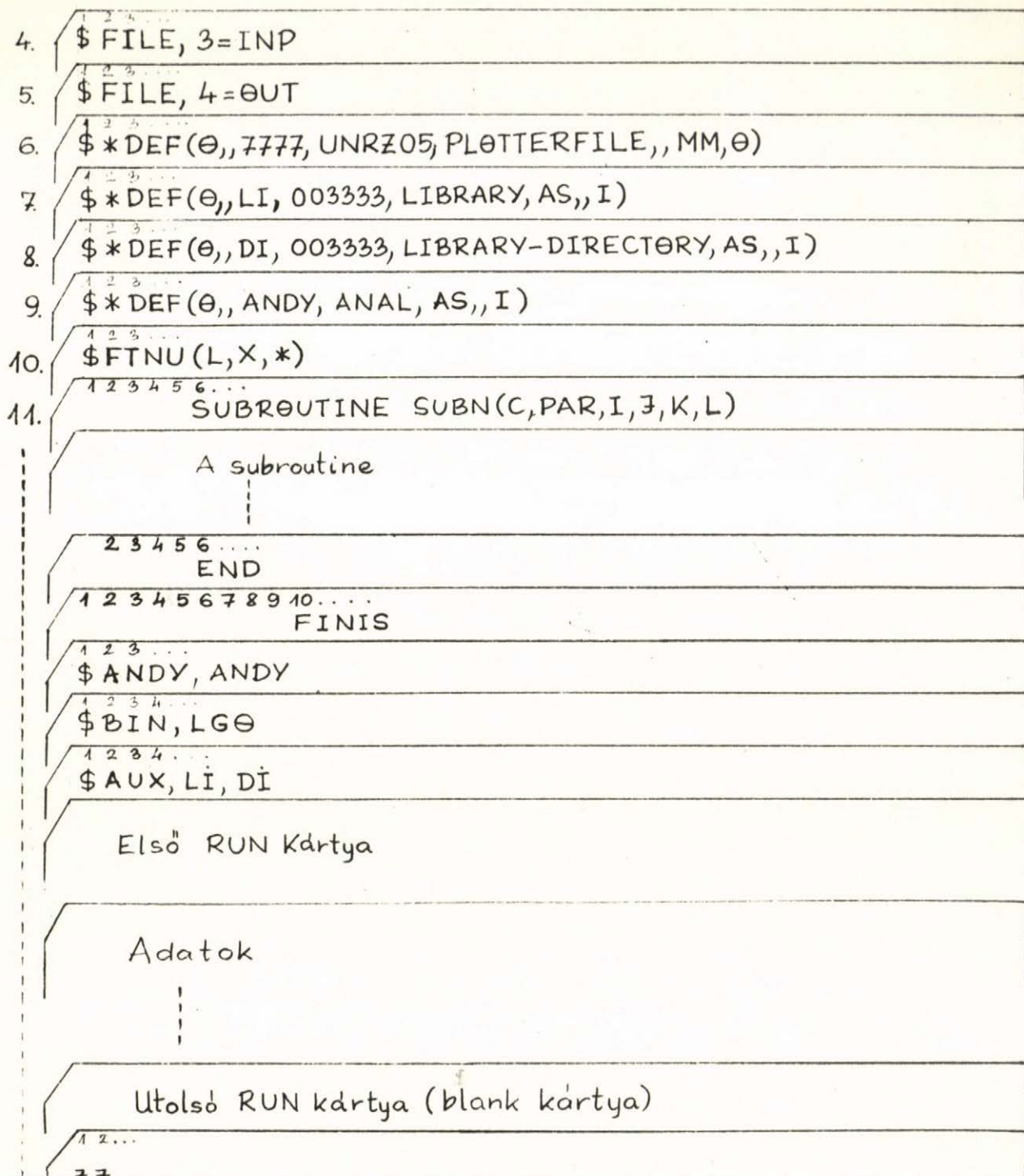
5.3. Példaként mutatjuk itt a $\sin(\omega t + \varphi)$ függvényt használó sinus jelgenerátort, melynek beépítéséhez a következő FORTRAN utasításokat kell kártyára lelyukasztatni:

```
SUBROUTINE SUB4(C,PAR,I,J,K,L)
  DIMENSION C(76), PAR(75,3)
  HI=PAR(I,1)* C(76)
  HI=HI+PAR(I,2)
  C(I)= SIN(HI)
  RETURN
END
```

A programban ω -át a PAR(I,1) -gyel, φ -t a PAR(I,2)-vel tekintjük megadottnak. A jelgenerátornak egyetlen bemenő adata az idő.

5.4. Ha a felhasználó szubrutinját FORTRAN forrásnyelven megírta és kártyára lelyukasztatta, akkor ezt a kártyacsomót a következőképpen kell elhelyezni az adatkártyacsomó vezérkártyái között:

```
1. $ JOB,
2. $ SCHED, CORE=76, SCR=2, 854=1, OUT=512, CLASS=I
3. $*DEF(0,XXX3,0003,PLOTTERFILE-HEADER,,,0)
```



77
88

Az ilyen módon összeállított kártyacsomó biztosítja, hogy az általa ini-
ciált feladat-futtatás időtartamára a SUBN speciális rutin is lefor-
ditva működésre készen áll.

6. Az ANDISIM hibajelzései

- 6.1. Amikor a felhasználó automatikus lépéshosszváltó integrálási módszerrel futtatja feladatát, az integrálási eljárás paramétereinek helytelen megválasztása, vagy a feladat instabil jellege következtében ugynevezett "erőszakolt végrehajtás"-ra kényszerülhet a szimulátor, melyet a "FORCED EVALUATION" felirattal jelez az eredmények nyomtatása közben. Automatikus lépéshosszváltó integrálásnál ugyanis mindig meg kell adni egy relatív hibakorlátot, hogy a kiindulási lépéshossz hány-szorosánál, ill. hányad részénél /EPSMAX/ nagyobb hiba esetén legyen lépéshossz felezés, továbbá, hogy a kiindulási lépéshossz hányad része /DELMIN/ az a minimális érték, amelynél tovább már nem szabad felezni. Előfordulhat, hogy vagy a relatív hibakorlátot túl kicsire, a minimum értéket pedig viszonylag nagyra állítja be a felhasználó, vagy az integrálási hiba oly gyorsan növekszik, hogy a beadott hibakorlát szerint a szimulátor programnak még tovább kellene feleznie a lépéshossz-szat, de a megadott minimumnál nem mehet tovább. Ilyenkor ezt a minimumértéket fogadja el új lépéshossznak /H1/ a szimulátor és ezt ki is írja. A számítás ezzel fut tovább. Ezt jelenti a "FORCED EVALUATION".
- 6.2. Ha véletlenül negatív blokkorszámot talál a paramétereket specifikáló kártyákon a szimulátor, akkor még a beolvasás közben leáll "ERROR HALT" hibaüzenettel.
- 6.3. Valahányszor az ANDISIM-be beépített automatikus szortoló /rendező/ rutin algebrai hurkot észlel az adott blokk-konfigurációban, még a számítás kezdete előtt "SORT FAILURE" hibaüzenettel leállítja a feladat futását.
- 6.4. Amennyiben a printeren kinyomtatandó sorok száma az ezret meghaladja, a futás a számítás vége előtt leáll a következő hibaüzenettel:

```
IAC = EO4  P = ..... INSTRUCTION=.....  
TASK SCOOP  
NAME = ..... P=..... STATUS=.....  
stb.
```

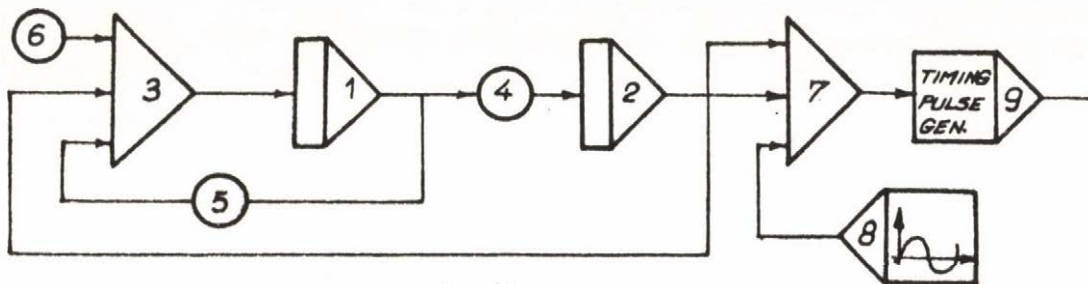

- 6.5. Ha a plotteren kirajzolandó blokkszám meghaladja az egy plotter output előíráshoz /egy RUN kártyához/ rendelkezésre álló 100 blokkot /1 blokk=1024 plotter lépés/, akkor a feladat futása automatikusan leáll.
- 6.6. A plotter rajzot specifikáló kártyán legfeljebb 1000 plotterlépést szabad megadni az ordinátatengely hosszára. Ennek túllépése a rajz elmaradását és az esetleges további RUN kártyák mellőzését vonja maga után.
- 6.7. Az ANDISIM egyáltalán nem ad output eredményt, ha a printelés lépésközét kisebbnek specifikálja a felhasználó az integrálási lépéshossznál.
- 6.8. Ha az ordináta skálafaktort $6.00E-14$ -nél kisebbnek adja meg a felhasználó egy "CAN NOT ROUND PLOTTER TICKS" feliratot és csupán x tengelyt ábrázoló plotterrajzot kap eredményül. Ennek oka, hogy az OUTPUT-ok értéke 0, így a skálafaktorozás lehetetlen.

Megjegyzés: A plotterrel kapcsolatos egyéb hibajelzések a "CDC 3300/FORTRAN Plotter rutinok használata" című kiadványban találhatók.

F Ü G G E L É K

1. Feladat

Az alábbi blokkvázlattal ábrázolt feladatban az idő impulzus generátor működését mutatjuk be.



3. ábra

Az adatkártyacsomó a következő kártyákból áll:

- | | |
|----|---|
| 1. | <div style="border: 1px solid black; padding: 2px;"> <div style="display: flex; justify-content: space-between; font-size: small;"> 1 2 3 4 5 80 </div> <div style="margin-top: 5px;">\$JOB, 22F345, PULS, 2, 1500, ,,</div> </div> |
| 2. | <div style="border: 1px solid black; padding: 2px;"> <div style="display: flex; justify-content: space-between; font-size: small;"> 1 2 3 4 5 80 </div> <div style="margin-top: 5px;">\$SCHED, CORE = 81, SCR = 2, CLASS = I, 854 = 1, OUT = 512</div> </div> |
| 3. | <div style="border: 1px solid black; padding: 2px;"> <div style="display: flex; justify-content: space-between; font-size: small;"> 1 2 3 4 5 80 </div> <div style="margin-top: 5px;">\$*DEF(Θ, , XXX3, 0003, PLOTTERFILE-HEADER, ,, Θ)</div> </div> |
| 4. | <div style="border: 1px solid black; padding: 2px;"> <div style="display: flex; justify-content: space-between; font-size: small;"> 1 2 3 4 5 80 </div> <div style="margin-top: 5px;">\$*DEF(Θ, , ANAL, 003333, ANAL, ,, I)</div> </div> |
| 5. | <div style="border: 1px solid black; padding: 2px;"> <div style="display: flex; justify-content: space-between; font-size: small;"> 1 2 3 4 5 80 </div> <div style="margin-top: 5px;">\$FILE, 3 = INP</div> </div> |
| 6. | <div style="border: 1px solid black; padding: 2px;"> <div style="display: flex; justify-content: space-between; font-size: small;"> 1 2 3 4 5 80 </div> <div style="margin-top: 5px;">\$FILE, 4 = OUT</div> </div> |

7.	1 2 3 4 5 80				
	\$*DEF(0,, 7777, UNRZ05. PLOTTERFILE,, MM,0)					
8.	1 2 3 4 5 80				
	\$*DEF(0,, ALIB, MASTER, LIBRARY, PL, *D0*)					
9.	1 2 3 4 5 80				
	\$*DEF(0,, ADIR, MASTER, LIBRARY-DIRECTORY, PL, *D0*)					
10.	1 2 3 4 5 80				
	\$ANAL, ANAL					
11.	1 2 3 4 5 80				
	\$AUX, ALIB, ADIR					
12.	1 2 3 4 5 6 7 8	... 80				
	11111000					
13.	1 2 3 4 ...	18 19 20	29 30	39 40	49 50	59 60 ... 80
		1	25	3		
14.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60 ... 80
		2	25	4		
15.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60 ... 80
		3	54	6	2	5
16.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60 ... 80
		4	23	1		
17.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60 ... 80
		5	23	1		
18.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60 ... 80
		6	34			
19.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60 ... 80
		7	54	2	2	8
20.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60 ... 80
		8	12			
21.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60 ... 80
		9	51	7		

22.	1 2 3 ...	blank kártya					... 80
23.	1 2 3 ...	19 20	35	50	65	...	80
		3	1.0	-1.0	-1.0		
24.	1 2 3 ...	19 20	35	50	65	...	80
		4	1.0				
25.	1 2 3 ...	19 20	35	50	65	...	80
		5	1.4				
26.	1 2 3 ...	19 20	35	50	65	...	80
		6	100.0				
27.	1 2 3 ...	19 20	35	50	65	...	80
		7	1.0	-1.0	1.0		
28.	1 2 3 ...	19 20	35	50	65	...	80
		8	12.56	4.0			
29.	1 2 3 ...	19 20	35	50	65	...	80
		9	0.05				
30.	1 2 3 ...	blank kártya					... 80
31.	1 2 3 4 5 6 ...	20	35	50	...	80	
		1	0.05	5.0	0.05		
32.	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	...	80				
		2	7	8	9		
33.	1 2 3 4 5 6 7 8	80				
		00000000					
34.	1 2 3	80				
		77					
		88					

Mint az első RUN kártyából kitűnik egyetlen futást igényeltünk plotter-használat nélkül.

A futás végén a sornyomtatón visszairatott adatok után az alábbi eredményeket kaptuk:

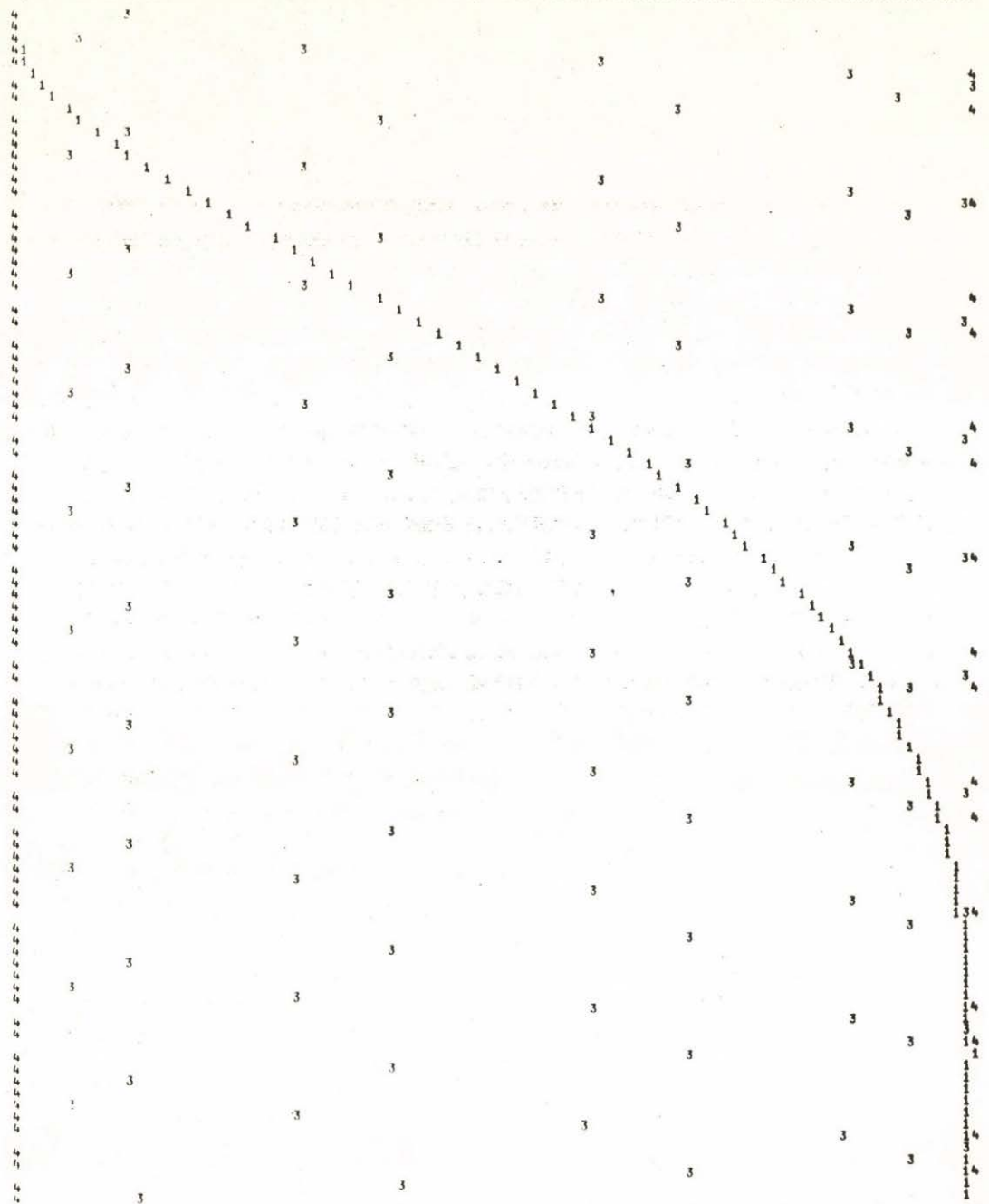
TIME	RKSTEP	OUTPUT 2	OUTPUT 7	OUTPUT 8	OUTPUT 9
0.00000	0.50000E-01	0.00000E+00	-0.75680	-0.75680	0.00000E+00
0.05000	0.50000E-01	0.12500	-0.99644	-0.99644	0.00000E+00
0.10000	0.50000E-01	0.48265	-0.85585	-0.85585	0.00000E+00
0.15000	0.50000E-01	1.0564	-0.38867	-0.38867	0.00000E+00
0.20000	0.50000E-01	1.8301	0.22682	0.22682	1.0000
0.25000	0.50000E-01	2.7885	0.75576	0.75576	0.00000E+00
0.30000	0.50000E-01	3.9168	0.99631	0.99631	1.0000
0.35000	0.50000E-01	5.2007	0.85667	0.85667	0.00000E+00
0.40000	0.50000E-01	6.6266	0.39013	0.39013	1.0000
0.45000	0.50000E-01	8.1815	-0.22527	-0.22527	0.00000E+00
0.50000	0.50000E-01	9.8528	-0.75472	-0.75472	0.00000E+00
.
.
.

PLOTTED OUTPUTS

2. BLOCKSYMBOL 1
3. BLOCKSYMBOL 2
4. BLOCKSYMBOL 3
5. BLOCKSYMBOL 4

TIME

0.0000
0.0500
0.1000
0.1500
0.2000
0.2500
0.3000
0.3500
0.4000
0.4500
0.5000
0.5500
0.6000
0.6500
0.7000
0.7500
0.8000
0.8500
0.9000
0.9500
1.0000
1.0500
1.1000
1.1500
1.2000
1.2500
1.3000
1.3500
1.4000
1.4500
1.5000
1.5500
1.6000
1.6500
1.7000
1.7500
1.8000
1.8500
1.9000
1.9500
2.0000
2.0500
2.1000
2.1500
2.2000
2.2500
2.3000
2.3500
2.4000
2.4500
2.5000
2.5500
2.6000
2.6500
2.7000
2.7500
2.8000
2.8500
2.9000
2.9500
3.0000
3.0500
3.1000
3.1500
3.2000
3.2500
3.3000
3.3500
3.4000
3.4500
3.5000
3.5500
3.6000
3.6500
3.7000
3.7500
3.8000
3.8500
3.9000
3.9500
4.0000
4.0500
4.1000
4.1500
4.2000
4.2500
4.3000
4.3500
4.4000
4.4500
4.5000
4.5500
4.6000
4.6500
4.7000
4.7500
4.8000
4.8500
4.9000
4.9500
5.0000



2.ELK	0.000000	10.5	20.9	31.4	41.8	52.3	62.8	73.2	83.7	94.1	105.
3.ELK	-0.016	-0.797	-0.598	-0.399	-0.199	-0.678E-04	0.199	0.398	0.598	0.797	0.996
4.ELK	-0.996	-0.797	-0.598	-0.399	-0.199	-0.678E-04	0.199	0.398	0.598	0.797	0.996
5.ELK	0.000E+00	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.800	0.900	1.00

LENGTH OF Y AXIS 1000.
LENGTH OF X AXIS 5000.

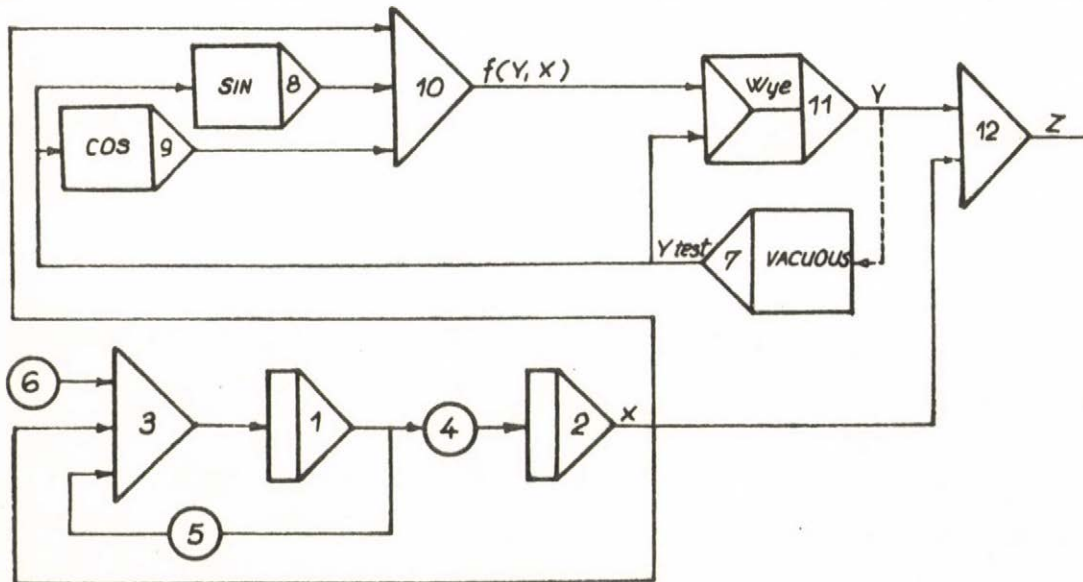
Y SCALE FACTORS FOR PLOTTED OUTPUTS
0.573487E+11 0.301492E+11 0.301492E+11 0.301492E+11

UNDEFINED OSI 7777

00006
SE-0000000000

2. Feladat

Az implicit függvény elemek együttes alkalmazását és funkcióját mutatja be konvergens algebrai hurok kezelésére az alábbi feladat:



4. ábra

Mielőtt az adatkártyacsomó összeállítására rátérnénk, rövid magyarázatot fűzünk a fenti konfigurációban az egyes elemek ill. elemcsoportok szerepéhez.

Az első hat blokk az X mennyiséget szolgáltatja az $Y=f(Y,X)$ implicit egyenlet megoldásához.

A 7. blokk $PAR(7,1)=0.13$ paraméterét mint az Y -ra adott kezdő becslést használja a szimulátor a számítás indításakor.

A 8., 9. és 10. blokkokból álló elemcsoport $f(Y,X)$ -re ad közelítő értéket.

A 11. blokk első, $PAR(11,1) = 0.001$ paramétere egy relatív hibakritériumot tartalmaz. Ha a 10. és 7. blokk kimeneti értékei közti relatív eltérés kisebb ennél a specifikált kritikus értéknél, akkor a szimulátor a 11. blokk kimeneti értékét egyenlőnek veszi a 10. blokk kimeneti értékével. Ha a 10. és a 7. blokk kimeneti értékei közti relatív eltérés nem kisebb a $PAR(11,1)$ -gyel megadott 0.001 értéknél, akkor a 11. blokk, a Wye elem a következőképpen működik:

- a.) a Wye elem második paramétere, $PAR(11,2) = 0.6$ segítségével egy újabb becslést szolgáltat Y-ra:

$$Y_{n+1} = (1 - PAR(11,2)) \cdot f(Y_n, X) + PAR(11,2) \cdot Y_n$$

- b.) ezt az Y_{n+1} értéket a 7. blokk újabb kimeneti értékéül fogadja el a szimulátor az előző Y_n érték helyett,
- c.) ezután visszatér a szimulátor a 8., 9. és 10. blokkok kimeneti értékeinek számításához az új Y értékkel.

Amíg az iteráció be nem fejeződik, az 1.-6. és a 12. blokk kimeneti értékét nem számítja ki újból a szimulátor.

Most rátérünk a feladat adatkártyacsomójának összeállítására:

	1 2 3 4 ...
1.	\$JOB, 26D112, IMPL, 2, 2000,,,
2.	\$SCHED, CORE=81, SCR=2, CLASS=1, 854=1, OUT=512
3.	\$*DEF(0,, XXX3, 0003, PLOTTERFILE-HEADER,, ,0)
4.	\$*DEF(0,, ANAL, 003333, ANAL,, ,I)
5.	\$FILE, 3=INP
6.	\$FILE, 4=OUT
7.	\$*DEF(0,, 7777, UNRZ05, PLOTTERFILE,, MM,0)
8.	\$*DEF(0,, ALIB, MASTER, LIBRARY, PL,*D0*)
9.	\$*DEF(0,, ADIR, MASTER, LIBRARY-DIRECTORY, PL,*D0*)
10.	\$ANAL, ANAL
11.	\$AUX, ALIB, ADIR

12.	1 2 3 4 5 6 7 8 80
	1 1 1 1 1 1 1 1	

13.	1 2 3 ...	19 20	29 30	39 40	49 50	59 60	... 80
		1	25	3			
14.		2	25	4			
15.		3	54	6	2	5	
16.		4	23	1			
17.		5	23				
18.		6	34				
19.		7	53				
20.		8	50	7			
21.		9	19	7			
22.		10	54	2	8	9	
23.		11	56	10	7		
24.		12	54	11	2		
25.	blank kártya						

26.	1 2 3 ...	19 20	35	50	65	... 80
		3	1.0	-1.0	-1.0	
27.		4	1.0			
28.		5	1.4			
29.		6	100.0			
30.		7	0.13			
31.		10	1.0	2.0	3.0	
32.		11	0.001	0.6		
33.		12	-1.0	1.0		
34.	blank kártya					

35.	1 2 3 4 5 6 20	25	50	...	80
	1	0.05	5.0	0.05	
36.	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	80		
	2	7	10	11	
37.	1 2 3 4 5 6 20	25	...	80	
	7777	1000.	600.		
38.	1 2 3 4 5 6 7 8	...	80		
	00000000				
39.	1 2 3	80		
	77				
	88				

Ennél a feladatnál egyszeri futtatást igényeltünk plotter használatával a bevezető RUN kártyán.

A sornyomtatón kiírt értékei a regisztrált kimeneteknek a következők voltak:

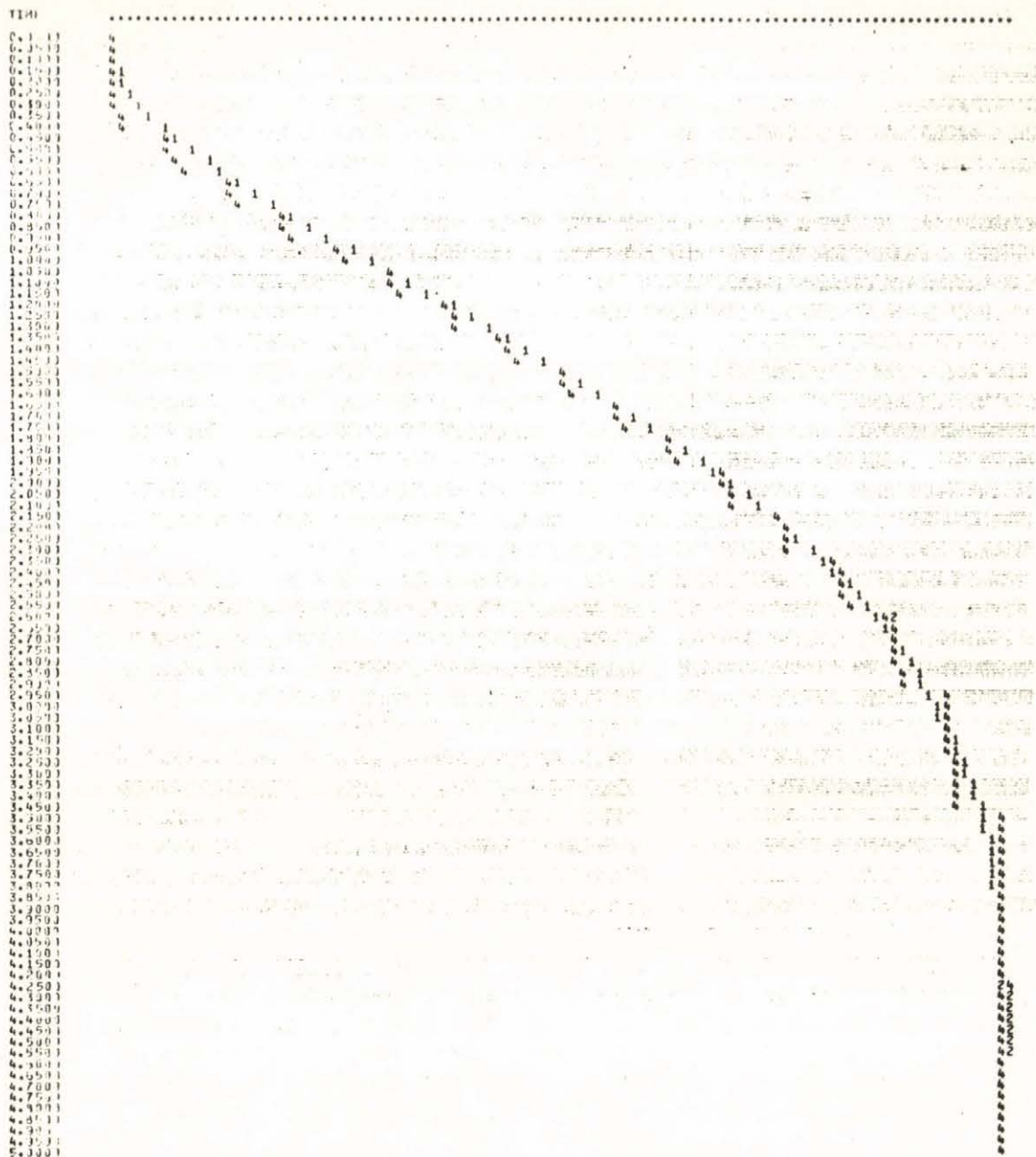
TIME	RKSTEP	OUTPUT 2	OUTPUT 7	OUTPUT 10	OUTPUT 11
0.00000	0.50000E-01	0.00000E+00	1.6758	1.6744	1.6744
0.05000	0.50000E-01	0.12500	1.7048	1.7064	1.7064
0.10000	0.50000E-01	0.48265	1.7886	1.7871	1.7871
0.15000	0.50000E-01	1.0564	1.9172	1.9189	1.9189
0.20000	0.50000E-01	1.8301	2.0878	2.0859	2.0859
0.25000	0.50000E-01	2.7885	2.2964	2.2941	2.2941
0.30000	0.50000E-01	3.9168	2.5476	2.5499	2.5499
0.35000	0.50000E-01	5.2007	2.8632	2.8658	2.8658
0.40000	0.50000E-01	6.6266	3.3198	3.3196	3.3196
0.45000	0.50000E-01	8.1815	8.3870	8.3795	8.3795
.
.
.

A printeren kirajzolt megfelelő görbék:

RAJZSZAM 1

PLOTTED OUTPUTS

2. BLOCKSYMBOL 1
7. BLOCKSYMBOL 2
10. BLOCKSYMBOL 3
11. BLOCKSYMBOL 4



2.BLK	9.010E+00	19.5	20.9	31.4	41.9	52.3	62.8	73.2	83.7	94.1	105.
7.BLK	1.65	11.9	22.0	32.1	42.3	52.4	62.5	72.7	82.8	93.0	103.
10.BLK	1.57	11.1	22.0	32.1	42.3	52.4	62.6	72.7	82.9	93.0	103.
11.BLK	1.57	11.9	22.5	32.1	42.3	52.4	62.6	72.7	82.9	93.0	103.

LENGTH OF X AXIS 1000.
LENGTH OF Y AXIS 500.

SCALE FACTORS FOR PLOTTER OUTPUTS
0.573437E+02 0.573437E+01 0.573437E+01 0.573437E+01

UNDEFINED 351 7770

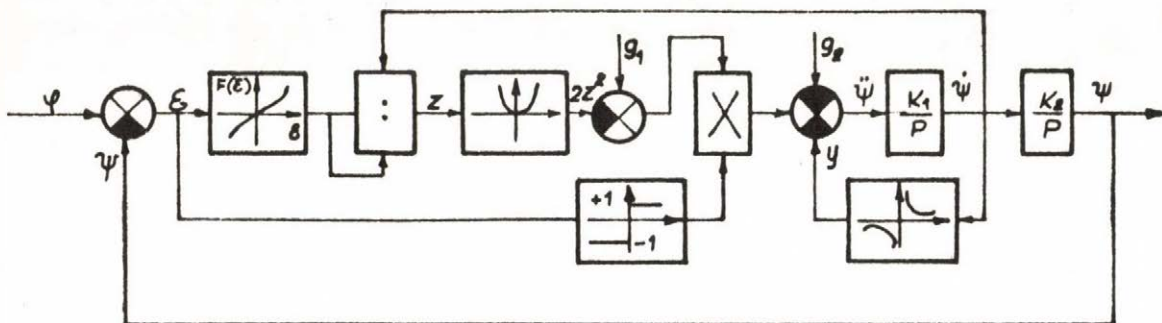
300000
SEQUENCE 0.0000

3. Feladat

Egy elektrohidraulikus léptetőmotor viselkedésének vizsgálata kapcsán most egyidejűleg több elem használatával ismerkedhet meg a felhasználó.

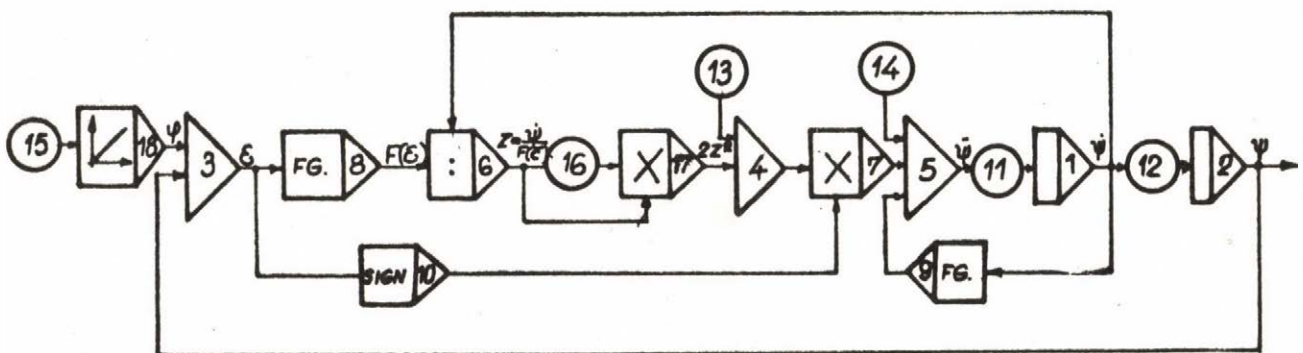
Itt kerül sor a paramétermódosítással történő újrafuttatásra is.

A feladat blokkvázlata a következő:



5. ábra

amelynek az alábbi elemekre lebontott működési vázlat felel meg:



6. ábra

A feladat futtatásához szükséges adatkártyacsomó a következő kártyákból állt:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1.																																\$J0B, 27U003, LEEPT, 5, 3000,,,																																																
2.																																\$SCHED, CORE=81, SCR=2, CLASS=I, 854=1, OUT=512																																																
3.																																*\$DEF(0,,XXX3,0003,PLOTTERFILE-HEADER,,0)																																																
4.																																*\$DEF(0,,ANAL,003333,ANAL,,I)																																																
5.																																\$FILE, 3=INP																																																
6.																																\$FILE, 4=OUT																																																
7.																																*\$DEF(0,,7777,UNRZ05,PLOTTERFILE,,MM,0)																																																
8.																																*\$DEF(0,,ALIB,MASTER,LIBRARY,PL,*D0*)																																																
9.																																*\$DEF(0,,ADIR,MASTER,LIBRARY-DIRECTORY,PL,*D0*)																																																
10.																																\$ANAL, ANAL																																																
11.																																\$AUX, ALIB, ADIR																																																

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
12.																																11111111																																																

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
13.																																1	25	11																																														
14.																																2	25	12																																														
15.																																3	54	18	2																																													
16.																																4	54	13	17																																													
17.																																5	54	14	7	9																																												
18.																																6	49	1	8																																													
19.																																7	55	4	10																																													
20.																																8	22	3																																														
21.																																9	22	1																																														
22.																																10	18	3																																														
23.																																11	23	5																																														
24.																																12	23	1																																														

	1 2 3 ...	19 20	29 30	39 40	49 50	59 60	... 80
25.		13	34				
26.		14	34				
27.		15	34				
28.		16	23	6			
29.		17	55	16	6		
30.		18	10	15			
31.	blank kártya						
32.	1 2 3 ...	10	20	30	40	50	60
33.	9	8	-12.6148	-4.9	-0.614	-0.000001	0.000001
34.		-100.0	-46.0	-16.0	-0.2	0.0	0.2
35.	4.9	12.6148					
36.	46.0	100.0					
37.	1 2 3 ...	10	20	30	40	50	60
38.	9	-4.0	-4.5	-5.0	-6.0	-9.9964	10.0
39.		-50.0	-30.0	-20.0	-10.0	-0.009	0.0
40.	5.0	4.5	4.0				
41.	20.0	30.0	50.0				
42.	1 2 3 4 ...	19 20	35	50	65	... 80	
43.		3	1.0	-1.0	0.0		
44.		4	1.0	-1.0	0.0		
45.		5	-1.0	1.0	-1.0		
46.		11	22.415	0.0	0.0		
47.		12	360.0	0.0	0.0		
48.		13	63.0	0.0	0.0		
49.		14	0.0	0.0	0.0		
50.		15	15.0	0.0	0.0		
51.		16	2.0	0.0	0.0		
52.	blank kártya						

52.	1 2 3 4 5 20 35 50 ... 80	3 0.0004 0.15 0.005
53.		1.0 0.1 0.1
54.	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 ... 80	3 8 6 1 2
55.	1 2 3 4 5 20 35 ... 80	7777 1000.0 600.0
56.	1 2 3 4 5 6 7 8 80	11111111
57.	1 2 3 ... 19 20 35 50 65 ... 80	11 44.83 0.0 0.0
58.		blank kártya
59.	1 2 3 4 5 6 7 8 80	00000000
60.	1 2 80	77 88

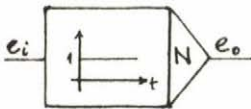
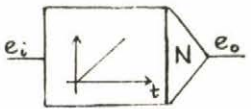
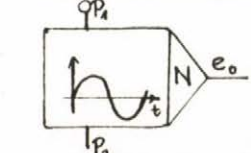
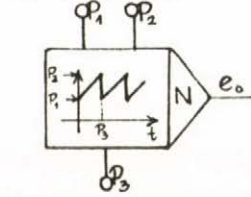


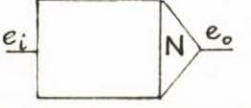

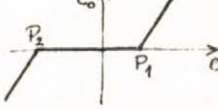
A feladat első futása során /ekkor a 11.blokk első paraméter $PAR(11,1) = 22.415$ volt/ az alábbi eredményeket kaptuk sornyomtatón:



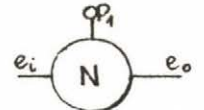

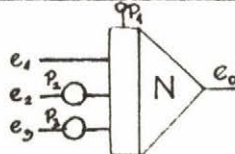

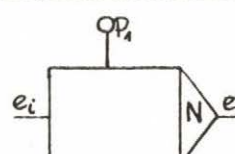
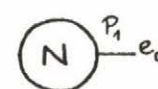
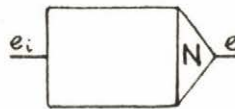
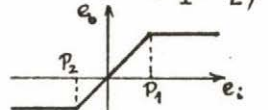
TIME	RKSTEP	OUTPUT 3	OUTPUT 8	OUTPUT 6	OUTPUT 1	OUTPUT 2
0.00000	0.10000E-03	0.00000E+00	0.10000E-05	0.00000E+00	0.00000E+00	0.00000E+00
0.00500	0.10000E-03	11.653	0.44507	5.4476	2.4246	3.3471
0.01000	0.10000E-03	7.9905	0.30275	5.4620	1.6536	7.0095
0.01500	0.10000E-03	5.5386	0.20746	5.3531	1.1106	9.4614
0.02000	0.10000E-03	3.8806	0.14303	5.2851	0.75593	11.119
0.02500	0.10000E-03	2.7473	0.98990E-01	5.2410	0.51881	12.253
0.03000	0.10000E-03	1.9674	0.68684E-01	5.2116	0.35795	13.033
0.03500	0.10000E-03	1.4284	0.47737E-01	5.1918	0.24784	13.572
0.03000	0.10000E-03	1.0547	0.33216E-01	5.1782	0.17200	13.945
0.04500	0.10000E-03	0.79520	0.23131E-01	5.1689	0.11956	14.205
0.05000	0.10000E-03	0.61470	0.16117E-01	5.1625	0.83202E-01	14.385
0.05500	0.10000E-03	0.48905	0.11234E-01	5.1580	0.57943E-01	14.511
0.06000	0.10000E-03	0.40152	0.78320E-02	5.1549	0.40374E-01	14.598
0.06500	0.50000E-04	0.34051	0.54615E-02	5.1528	0.28142E-01	14.659
:	:	:	:	:	:	:

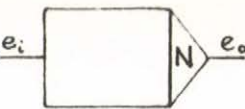
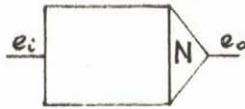
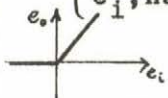
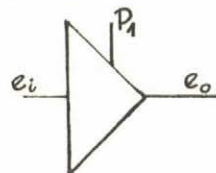
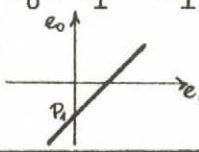


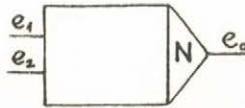

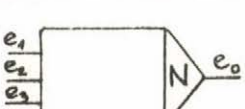
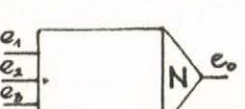
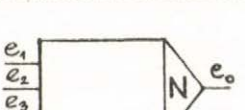
A feladat második futása során/PAR(11,1) = 44.83 paraméterértékkel/ a következők voltak az eredmények:

TIME	RKSTEP	OUTPUT 3	OUTPUT 8	OUTPUT 6	OUTPUT 1	OUTPUT 2
0.00000	0.10000E-03	0.00000E+00	0.10000E-05	0.00000E+00	0.00000E+00	0.00000E+00
0.00500	0.10000E-03	11.034	0.42103	5.3785	2.2645	3.9656
0.01000	0.10000E-03	7.6582	0.28983	5.3027	1.5369	7.3418
0.01500	0.10000E-03	5.3565	0.20039	5.2527	1.0526	9.6435
0.02000	0.10000E-03	3.7754	0.13894	5.2196	0.72522	11.225
0.02500	0.10000E-03	2.6838	0.96524E-01	5.1972	0.50166	12.316
0.03000	0.10000E-03	1.9277	0.67142E-01	5.1820	0.34793	13.072
0.03500	0.10000E-03	1.4029	0.46746E-01	5.1715	0.24175	13.597
0.04000	0.10000E-03	1.0380	0.32566E-01	5.1643	0.16818	13.962
0.04500	0.10000E-03	0.78402	0.22696E-01	5.1593	0.11710	14.216
0.05000	0.10000E-03	0.60714	0.15823E-01	5.1558	0.81579E-01	14.393
0.05500	0.10000E-03	0.48389	0.11033E-01	5.1547	0.56872E-01	14.516
.
.
.

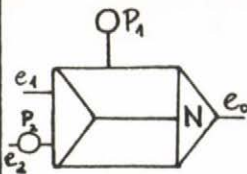
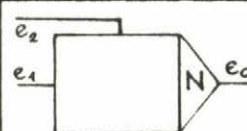
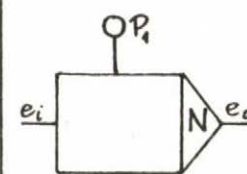
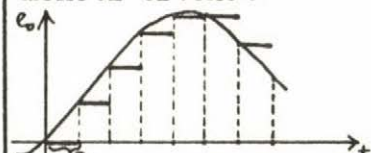
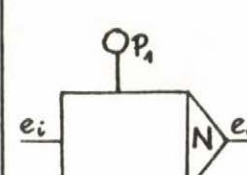
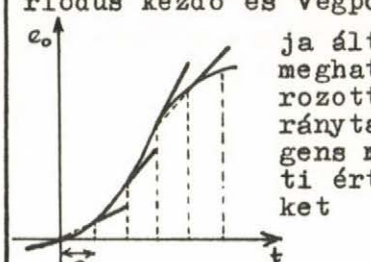
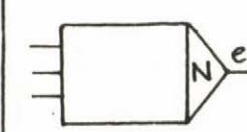
I. Táblázat
Az ANDISIM elemkészlete

Az elem neve	Tipus jelző száma	Jelölése	Műveleti funkciója	Bemeneti paraméte- rei
Egységugrás jelgenerátor /Step function/	10		$e_o = \begin{cases} 1., & \text{ha nincs } e_i \text{ be-} \\ & \text{menet} \\ e_i, & \text{ha van } e_i \text{ beme-} \\ & \text{net} \end{cases}$	
Sebességugrás jelgenerátor /Ramp function/	11		$e_o = \begin{cases} t, & \text{ha nincs } e_i \text{ be-} \\ & \text{menet} \\ e_i \cdot t, & \text{ha van } e_i \\ & \text{bemenet} \end{cases}$	
Sinus-függvény jelgenerátor /Sine function/	12		$e_o = \sin(P_1 \cdot t + P_2),$ ahol $P_1 = \omega$ $P_2 = \varphi$	$P_1, P_2,$
Fűrészfog jelgenerátor /Saw-tooth wave/	13		$e_o = \begin{cases} P_1, & \text{ha } t=0.0 \\ \frac{P_2 - P_1}{P_3} (t - nP_3), & \\ P_2, & \text{ha } t=(n+1) \cdot P_3 \end{cases}$ ahol $n \geq 0$ egész	P_1, P_2 P_3
Arcus tangens /Archtangent/	17		$e_o = \arctan(e_i)$	
Signum /Bang-bang/	18		$e_o = \begin{cases} -1., & \text{ha } e_i < 0. \\ 0., & \text{ha } e_i = 0. \\ +1., & \text{ha } e_i > 0. \end{cases}$	
Cosinus /Cosine/	19		$e_o = \cos(e_i),$ ahol e_i radiánokban értendő	
Holt zóna /Dead space/	20		$e_o = \begin{cases} 0, & \text{ha } e_i = 0 \\ \max(0, e_i - P_1), & \text{ha } e_i > 0 \\ \min(0, e_i - P_2), & \text{ha } e_i < 0 \end{cases}$ 	P_1, P_2

Az elem neve	Tipus jelző száma	Jelölése	Műveleti funkciója	Bemeneti paraméte- rei
Exponenciális /Exponential/	21		$e_o = \exp(e_i)$	
Függvénygenerátor /Function generátor/	22		$e_o = f(e_i)$ /lineáris interpoláció/	
Erősítő /Gain/	23		$e_o = P_1 \cdot e_i$	P_1
Négyzetgyök /Half power/	24		$e_o = \sqrt{e_i}$	
Integrátor /Integrator/	25		$e_o = P_1 + \int (e_1 + e_2 P_2 + e_3 P_3) dt$	P_1, P_2, P_3
Jelfordító /Signe inverter/	32		$e_o = -e_i$ /egy bemenet van csak/	
Véletlenszám ge- nerátor /Jitter/	33		-1. és 1. közötti vagy 0. és 1. közötti véletlen számot állít elő; ha $P_1=0$, akkor (0., 1.)-ben ha $P_1=-1$, akkor (-1., 1.)-ben	P_1
Konstans /Constant/	34		$e_o = P$	P_1
Korlátozó /Limiter/	35		$e_o = \begin{cases} \min(e_i, P_1) & \text{ha } e_i \geq 0 \\ \max(e_i, P_2) & \text{ha } e_i < 0 \end{cases}$ 	P_1, P_2

Az elem neve	Tipus jelző száma	Jelölése	Műveleti funkciója	Bemeneti paramé- tere
Abszolútérték /Magnitude/	36		$e_o = \text{abs}(e_i)$	
Negatív vágó /Negative clipper/	37		$e_o = \begin{cases} 0, & \text{ha } e_i \leq 0 \\ e_i, & \text{ha } e_i > 0 \end{cases}$ 	
Eltolás /Offset/	38		$e_o = e_i - P_1$ 	P_1
Pozitív vágó /Positive clipper/	39		$e_o = \begin{cases} e_i, & \text{ha } e_i < 0 \\ 0, & \text{ha } e_i \geq 0 \end{cases}$ 	
Összehasonlító /Quit/	40		ha $e_1 > e_2$, befejeződik a futás	
Kétállású relé /Relay/	41		$e_o = \begin{cases} e_2, & \text{ha } e_1 \geq 0 \\ e_3, & \text{ha } e_1 < 0 \end{cases}$	
Logikai "ÉS" kapu /.AND./	42		$e_o = \begin{cases} e_3, & \text{ha } e_1 \cdot e_2 \neq 0 \text{ és } e_3 \neq 0 \\ 1, & \text{ha } e_1 \cdot e_2 \neq 0 \text{ és } e_3 = 0 \\ 0, & \text{egyébként} \end{cases}$	
Logikai "VAGY" /.OR./	43		$e_o = \begin{cases} e_3, & \text{ha } e_1^2 + e_2^2 \neq 0 \text{ és } e_3 \neq 0 \\ 1, & \text{ha } e_1^2 + e_2^2 \neq 0 \text{ és } e_3 = 0 \\ 0, & \text{egyébként} \end{cases}$	
Logikai kizáró "VAGY" /.EOR./	44		$e_o = \begin{cases} e_3, & \text{ha } e_1 \cdot e_2 = 0, e_3 \neq 0 \text{ és } e_1^2 + e_2^2 \neq 0 \\ 1, & \text{ha } e_1 \cdot e_2 = 0, e_1^2 + e_2^2 \neq 0 \text{ és } e_3 = 0 \\ 0, & \text{egyébként} \end{cases}$	

Az elem neve	Tipus jelző száma	Jelölése	Műveleti funkciója	Bemeneti paramétere
Logikai azonosság /.IDENT./	45		$e_0 = \begin{cases} 0, & \text{ha } e_1 \cdot e_2 = 0 \\ & \text{és } e_1^2 + e_2^2 \neq 0 \\ e_3, & \text{ha } e_1 \cdot e_2 \neq 0, e_1^2 + e_2^2 = 0 \\ & \text{és } e_3 \neq 0 \\ 1, & \text{ha } e_1 \cdot e_2 \neq 0, e_1^2 + e_2^2 = 0 \\ & \text{és } e_3 = 0 \end{cases}$	
Osztó /Divider/	49		$e_0 = e_1 / e_2$	
Sinus /Sine/	50		$e_0 = \sin(e_i)$ ahol e_i radiánokban értendő	
Idő impulzus ge- nerátor /Timing pulse ge- nerator/	51		Csak $e_i \geq 0$ -nál ad impulzust. $e_i > 0$ esetén P_1 periódussal 	P_1
Késleltető /Unit delay/	52		$e_0 = e_i \left(t - \frac{\Delta t}{2} \right)$	
Implicit függ- vény I. / Vacuous/	53		$t=0.0$ -nál az $Y=f(Y,X)$ függvényre adja meg az $e_0=Y_0=P_1$ kezdeti értéket; $t>0.0$ -nál az Implicit fv.II állítja elő az e_0 kimeneti értéket az Implicit fv.I-re is.	P_1
Súlyozott összeg- ző /Weighted summer/	54		$e_0 = P_1 e_1 + P_2 e_2 + P_3 e_3$	P_1, P_2 P_3
Szorzó /Multiplier/	55		$e_0 = e_1 \cdot e_2$	

Az elem neve	Tipus jelző száma	Jelölése	Műveleti funkciója	Bemeneti paramétere
Implicit függvény II. /Wye/	56		Ha $\frac{ e_2 - e_1 }{ e_1 } \leq P_1$, akkor $e_o = e_1$. Ha $\frac{ e_2 - e_1 }{ e_1 } > P_1$, akkor $e'_2 = (1.0 - P_2)e_1 + P_2 \cdot e_2$, ahol e'_2 az Implicit fv I. új e_o -ja	P_1, P_2
Mintavételező / Sampler/	57		$e_o = \begin{cases} e_1, & \text{ha } e_2 \geq 0 \\ 0.0, & \text{ha } e_2 < 0 \end{cases}$	
Nullandrendű tartó /Zero-order hold/	58		P_1 periódusidőn keresztül tartja a periódus elejéhez tartozó e_i bemeneti értéket 	P_1
Elsőrendű tartó /First order hold/	59		P_1 periódusidőn keresztül tartja az előző periódus kezdő és végpontja által meghatározott iránytangens menti értéket 	P_1
Speciális felhasználói elemek 1-9 /Special 1-9 /	01 02 . . . 09		A felhasználó által irt tetszőleges rutinok	

A TANULMÁNYOK sorozatban eddig megjelentek:

- 1/1973 Pásztor Katalin: Módszerek Boole-függvények minimális vagy nem redundáns, { \wedge , \vee , \neg } vagy {NOR} vagy {NAND} bázisbeli, zárójeles vagy zárójel nélküli formuláinak előállítására
- 2/1973 Вашкеви Иштван: Расчленение многосвязных промышленных процессов с помощью вычислительной машины
- 3/1973 Ádám György: A számítógépipar helyzete 1972 második felében
- 4/1973 Bányász Csilla: Identification in the presence of drift
- 5/1973* Gyürki J.-Laufer J.-Girnt M.-Somló J.: Optimalizáló adaptív szerszámgépirányítási rendszerek

*-gal jelölt kivételével a TANULMÁNYOK megrendelhetők az Intézet Könyvtáránál /Budapest, I.Uri u. 49./



Jelen felhasználói kézikönyv a 2.7.3
"Nagy rendszerek dinamikájának vizs-
gálati módszerei" c. intézeti alapku-
tatási téma keretében készült.

